Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computer Communications 29 (2006) 2413–2432

computer communications

www.elsevier.com/locate/comcom

Performance evaluation of subflow capable SCTP ☆,☆☆

Jianping Zou [a], M. Ümit Uyar [a], Mariusz A. Fecko [b,*], Sunil Samtani [b]

[a] *The City College and the Graduate Center of the City University of New York, USA*

[b] *Applied Research Area, Telcordia Technologies, Inc., Piscataway, NJ 08854, USA*

Received 25 February 2005; received in revised form 7 November 2005; accepted 6 December 2005
Available online 24 January 2006

Abstract

With its new features such as multi-homing, multi-streaming, and enhanced security, the Stream Control Transmission Protocol (SCTP) has become a promising candidate to join UDP and TCP as a general-purpose transport layer protocol. Multiple streams in an SCTP association provide an aggregation mechanism to accommodate heterogeneous objects, which belong to the same application but may require different types of QoS from the network. However, the current SCTP specification lacks an internal mechanism to support the preferential treatment among its streams. Our earlier work introduced the concept of grouping SCTP streams into subflows based on their required QoS. We proposed to modify the current SCTP to implement subflows, each with its own flow and congestion mechanism to prevent the so-called *false sharing*. In this paper, performance evaluation of subflow capable SCTP (SF-SCTP) is studied through a set of extensive simulation experiments. The results show that the proposed SF-SCTP design is able to support QoS among the SCTP streams and that false sharing is avoided. The results also reveal SF-SCTP's significant benefits in improving the utilization of a bottleneck network.

Keywords: SCTP; Analytic model; Diff-Serv; QoS; False sharing; Congestion control

1. Introduction

The Stream Control Transmission Protocol (SCTP) [24,26] is a reliable, message-oriented transport protocol operating on top of potentially unreliable connectionless networks such as IP. Designed to overcome the shortcomings and limitations of TCP, SCTP is more suitable for applications requiring additional performance and reliability due to its new services such as multi-homing, multi-streaming, message boundary preservation, alleviated head-of-line blocking, and enhanced security features.

Using independent streams in a single association, SCTP decouples the reliable data transfer from the strict order-of-transmission data delivery. The messages from the application layer are assigned to different streams according to the requirements of an SCTP user. Since ordered delivery is only needed within each stream, if required, SCTP is able to reduce the unnecessary head-of-line blocking between different streams. Thanks to multi-streaming, SCTP is equipped with an internal mechanism to support transmission of several objects concurrently. For example, the HTTP protocol using SCTP as the transport layer could load a web page with multiple objects (e.g., JPG, voice, text, and video) by opening only one SCTP association instead of several TCP connections.

The current SCTP employs a TCP-like congestion control mechanism at the association level, which means the streams carrying different objects are treated equally according to the same congestion state information.

☆ Prepared through collaborative participation in the Communications & Networks Consortium sponsored by the U.S. Army Research Lab under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

☆☆ © 2005 Telcordia Technologies, Inc. and the City University of New York.

* Corresponding author.
E-mail address: mfecko@research.telcordia.com (M.A. Fecko).

0140-3664/$ - see front matter © 2006 Telcordia Technologies Inc. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2005.12.008

Assuming that SCTP packets could only bundle messages from the streams requesting the same type of QoS and that SCTP could mark its packets with Differentiated Services Codepoints (DSCPs) so that the network will treat those packets differently (in terms of packet loss rate, delay, etc.), the shared congestion information could result in the so-called *false sharing* [3]. False sharing happens when flows sharing the congestion information do not actually share the same network bottleneck, which could worsen the overall performance of SCTP association. Since the transport layer tries to moderate the performance for all streams, in the case where streams have different packet drop rates, the performance of the higher priority streams is penalized due to the shared information from the lower priority streams. When streams also experience different round trip times (RTTs), false sharing has catastrophic effects on SCTP performance in that all streams, regardless of their priorities, are penalized due to unnecessary Fast Retransmits or time-outs.

Our earlier work introduced the concept of grouping streams into subflows to support preferential treatment of the SCTP streams [32–34]. Each subflow includes streams with the same type of QoS requirement from the network. Depending on the number of QoS types requested, an SCTP association may have one or more subflows. We also proposed that each subflow be an independent unit with respect to flow and congestion control. In this modified subflow capable SCTP (SF-SCTP), each subflow implements its own flow and congestion control by using the original SCTP mechanisms at the subflow level. Since flow and congestion control are only shared among the streams belonging to the same subflow, SF-SCTP design inherently avoids false sharing.

The first set of simulations presented in this paper is conducted in an emulated Diff-Serv network where flows with different QoS markings are mapped onto different paths with distinct packet drop rates or RTTs. The simulation results show that our SF-SCTP design avoids false sharing and is able to provide preferential treatment to its streams. The results also confirm the accuracy of the analytic models derived in Refs. [31–34].

To better understand the congestion behavior of SF-SCTP, we conduct various experiments under a single bottleneck network with different network configurations and background traffic. The simulation results show that SF-SCTP with multiple subflows improves the bottleneck network utilization at the cost of high packet drop rate. We also find out that SF-SCTP is more effective in improving the utilization of a network that implements Random Early Detection (RED) [9] queue mechanism other than Drop-Tail, due to SF-SCTP's more consistent resilience to packet losses in a RED queue.

When competing against background traffic, SF-SCTP with multiple subflows offers greater benefits than the original SCTP because it seizes more bandwidth by adapting faster to network traffic changes. However, the bandwidth improvement of SF-SCTP comes partially at the expense of the background traffic. Our simulation results show that a bottleneck network's packet drop rate increases with the number of SF-SCTP subflows, directly hurting the throughput of background traffic. This aggressiveness problem of SF-SCTP can be alleviated by adding fractional congestion control into the design [31,34].

Section 2 of this paper outlines the techniques to improve the performance of transport layer. The design overview of SF-SCTP is presented in Section 3. Analytic models for the different SCTP implementations are presented in Section 4. The results of ns-2 experiments are discussed in Section 5.

2. Related work

Several techniques to improve the performance and QoS capabilities at the transport layer have been proposed [1–4,7,8,10,15,27].

In Ref. [4], the authors present an end-system architecture centered around a Congestion Manager (CM) that ensures proper congestion behavior and allows applications to easily adapt to network congestion. The CM is a middle layer responsible for congestion control of all TCP connections between TCP and IP layers. Independent multiple TCP connections cooperate rather than compete with each other. This framework integrates congestion management across all applications and transport protocols. The CM maintains congestion parameters and exposes an API to enable applications to learn about network characteristics, pass information to the CM, and schedule data transmissions. This special centralized congestion control scheme allows for multiple streams to share a network path while avoiding the false sharing problem.

Ref. [3] has investigated the origin and impact of false sharing on TCP performance. False sharing occurs in networks with QoS enhancements where a flow classifier segregates flows into different queues, or in networks with path diversity where different flows to the same destination are routed differently [3]. Their simulation results show that faster applications are heavily penalized as a result of false sharing. Without a separate flow and congestion control for each subflow in SF-SCTP, the benefits of a network supporting QoS might be forfeited due to transport layer's unawareness of QoS and inability to avoid false sharing.

In Ref. [10], aggregate TCP flows are used to improve network utilization. To address the unfairness of aggregate TCP flows towards a single competing TCP flow, the concept of fractional congestion control is introduced in Refs. [11,12]. Compared to a single TCP flow, the aggressiveness of multiple parallel TCP flows is due to a faster congestion window growth rate and a larger resilience to loss. Parallel TCP flows compete unfairly towards a single flow in that they open their congestion windows n (the number of flows) times faster. Parallel flows absorb the packet losses over the affected flows while allowing the rest of the flows to continue normal operation. For high speed networks where packet losses exclusively result in Fast Retransmits,